REMARKS

The Examiner rejected Claims 1-38 under 35 U.S.C. §101 as lacking patentable utility. The examiner stated:

> "It is not clear what is the utility of the processor, of the method executed in the processor, and of the program product. It is not clear what objective is served by placing the plurality of pointers in the stack and removing before it is known if a sequence of microinstructions pointed by the pointers is valid."

Applicant contends that this rejection is improper, since there is no statutory basis for the examiner to require that the claims recite the utility of the invention. Nonetheless, Applicant's claims clearly recite a utility as in "A processor, the processor implemented as a three way superscalar, pipelined architecture, the processor comprising:" as in claim 1 and the specification clearly discloses utility for this invention, namely in a processor or a method executed in a processor, and a program product. This utility is described in the specification at various places including page 5, line 3 through page 6, line 18.

For example, placing the plurality of pointers in the stack and removing before it is known if a sequence of microinstructions pointed by the pointers is valid provides a mechanism for improving the performance of microcode (µcode) execution [page 6, lines 3-4]. Microcode is programming that is ordinarily not program-addressable but, unlike hardwired logic, is capable of being modified. The microinstruction pointer (µIP) stack 100 provides a lower-overhead ability to jump to various subroutines and use "assists" to efficiently accomplish µcode functions. The µIP stack 100 has significant performance and µcode efficiency implications that permeate numerous instructions. For example, use of the µIP stack 100 improves performance by removing indirect µcode jumps and allows µcode to share routines more easily by removing subroutine penalties. By removing subroutine penalties, the µIP stack 100 allows µcode to share routines more easily [page 5, line 5-18].

Applicant submits that the claimed invention has utility and that Applicant's specification clearly points out the utility. Accordingly, Claims 1-38 are proper under 35 U.S.C. §101.

The examiner rejected claims 1-38 under 35 U.S.C. §112, first paragraph, as failing to comply with the enablement requirement.

The examiner stated: "Applicant failed to adequately describe a processor implementing a method of placing and removing a plurality of pointers associated with a common macroinstruction from an-out-of-order microinstructions pointer stack, and it would require an ordinary skill in the art undue experimentation to develop the processor and the method."

Applicant contends that the claims as presented are enabled by Applicant's specification, as originally filed. However, in order to advance prosecution, Applicant has clarified the claims to recite:" a common instruction that is decoded into a plurality of micro ops," since "macroinstruction" is too limiting. Applicant describes that:

> The front end 16 supplies instructions in program order to the out of order execution core 18 that has very high execution bandwidth and can execute basic integer operations with one-half clock cycle latency. The front end 16 fetches and decodes instructions into simple operations called micro-ops (μops). [Specification, page 2, lines 14-19].
>
> The out of order execution core 18 includes an out of order microinstruction pointer (IP) stack 100. In general, a stack is a data area or buffer used for storing requests that need to be handled. ... The stack 100 is implemented in a microcode environment. This allows fast subroutine returns in microcode. It also allows fast assist returns in microcode. [Specification, page 5, lines 3-13].
>
> The μIP stack 100 is an out-of-order stack where values are placed on the stack and removed from the stack before it is known if the sequence of operations were valid. Thus, a set of recovery mechanisms is required to correct a sequence of operations that are later determined to be incorrect. The process of manipulating the stack (PUSH, POP, etc.) is very different from a traditional macroinstruction stack because of the out-of-order nature of the stack 100. [Specification, page 5, line 18 to page 6, line 2].

The Examiner requested an explanation regarding the disclosure of the operations of placing and removing a plurality of pointers associated with a common macroinstruction. Thus, given the excerpts above and in view of Applicant's discussion on page 2, lines 14-19, that, while the front end 16 supplies instructions in program order to the out of order execution core 18, the front end 16 fetches and decodes instructions into simple operations called micro-ops

(μops), one of ordinary skill would understand that Applicant describes a processor implementing a method that places and removes a plurality of pointers associated with a common instruction from an-out-of-order microinstructions pointer stack.

Applicant notes that specification discloses the operations of the pointers. Examples are described on pages 14-27. If the examiner requires further explanation, the applicant respectfully requests the examiner to explain precisely what information is requested.

The Examiner rejected Claims 1-38 under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention:

> **(a) Claim 1, it is not clear what is out-of-order with respect to what.**
> **(b) Claim 1, it is not clear whether the plurality of pointers are placed concurrently and removed concurrently.**
> **(c) Claim 1, it is not clear whether microcode and microinstruction are used synonymously.**
> **(d) Claims 8 and 32, the same as (a) to (c), and it is not clear what is meant by "executing microcode addressed by (through) pointers stored in the ... stack" and what are the purpose and the result of the execution.**

The term "out-of-order" is permissible under 35 U.S.C. §112 since it is a commonly used and understood term in computer science. Applicant's specification clearly discloses: "The core 18 executes instructions out of order enabling the processor 10 to reorder instructions so that if one μop is delayed while waiting for data or a contended execution resource, other μops that are later in a program order may execute before the delayed μops." [page 4, lines 7-11]. Moreover, the specification, e.g., FIG. 1 and accompanying description pages 5-6, describe the out-of-order μIP stack in sufficient detail such that those skilled in the art would understand the use of "out-of-order." "Out-of-order" as applied to the microinstruction pointer, relates to how that stack differs from traditional microinstruction pointer stacks. In the out-of-order μIP stack, "values are placed on the stack and removed from the stack before it is known if the sequence of operations were valid." [Page 5, lines 18-20].

The specification, e.g., FIG. 4 and accompanying description pages 14-27 (includes a number of detailed examples), describe the placement and removal of the pointers in sufficient detail such that those skilled in the art would understand the use of the pointers.

The term "microcode" and "microinstruction" are permissible under 35 U.S.C. §112 since they are commonly used terms in computer science. It is commonly understood that microcode is comprised of microinstructions. The specification, e.g., FIG. 1 and accompanying description pages 6, describe the microcode and microinstruction in sufficient detail such that those skilled in the art would understand the use of "microcode" and "microinstruction."

Since Claim 1 is written to one of ordinary skill in the art, it is incumbent upon the examiner to show why one of ordinary skill in the art would not understand what is claimed. Accordingly, claim 1 is proper under 35 U.S.C. §112.

The phrase "executing microcode addressed by pointers stored in an out-of-order microinstruction pointer (μIP) stack" describes the method executed in the processor. Since the claim is written to one of ordinary skill in the art, it is incumbent upon the examiner to show why one of ordinary skill in the art would not understand what is claimed. Accordingly, claims 8 and 32 are proper under 35 U.S.C. §112.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue or comment does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 10/5/06

Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21399916.doc